**Problem Set 1 (Beginner)**

1. mtcars is a dataframe, I used the class() function to find out
2. precip is a 70 element vector with only $names as an attribute, to figure this out I initially tried using is.vector(precip) and str(precip), but then realized that it returns TRUE for list(). I then created a primitive function to test three things.

> is2 <- function(x)

+ {

+ c(is.vector(x), is.atomic(x), is.array(x))

+ }

> is2(precip)

[1] “TRUE” “TRUE” “FALSE”

The function of the first two elements is to determine through logic whether precip is a vector, and if so, is it an atomic vector and not a list? Another way to go about that might be to use is.recursive(x) to directly query if it is a list. The third element double checks that precip is not, in fact, a one dimensional array.

1. as.matrix(trees)
2. Atlanta (print(precip[14]))
3. > all <- list(“a”=precip, “b”=trees, “c”=mtcars)
4. Yes it does, specifically it is a double-precision floating point vector which all numeric data is stored as in R.

> typeof(precip)

[1] “double”

1. mtcars[2,7]; mtcars[2,][[7]]; mtcars[,7][[2]]; mtcars$qsec[[2]]
2. precip[c(“Juneau”, “Phoenix”, “Sacramento”)] <- c(23, 46, 12)
3. No. I used a nested ifelse function with logical operators to compare Girth and Volume, if there were any Girth values greater than its corresponding Volume it would return a 1 in the vector, and if not a 0.

> ifelse(trees$Girth>trees$Volume, 1, ifelse(trees$Girth<trees$Volume, 0, NA))

[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1. 2356.633

**Problem Set 2 (Intermediate)**

**Section 1**

1. The replace= argument of the sample() function indicate to R whether to sample with replacement (TRUE) or without (FALSE)
2. ifelse(MyMatrix==FALSE, 0, ifelse(MyMatrix==TRUE, 1, NA))
3. I formed a function, utilizing the previous ifelse statement. This works especially well because MyMatrix is already logical and need not be coerced.

> test <- function(x) {

+ MyMatrix=ifelse(MyMatrix==TRUE,1,ifelse(MyMatrix==FALSE,0,NA))

+ augMatrix=rowSums(MyMatrix)

+ matrix(data=ifelse(augMatrix==12,TRUE,ifelse(augMatrix!=12,FALSE,NA)))

+ }

> test(MyMatrix)

[,1]

[1,] FALSE

[2,] FALSE

[3,] FALSE

[4,] FALSE

[5,] FALSE

[6,] FALSE

[7,] FALSE

[8,] FALSE

**Section 2**

> length(which(MyMatrix==7))

[1] 16

> colSums(MyMatrix)

[1] 51 37 45 58 47 54 51 46 53 42 59 42

> apply(MyMatrix, 2, prod)

[1] 290304 46656 123480 3402000 340200 1080000 1975680 79380 987840

[10] 134400 4838400 72000

1. ifelse(MyMatrix==10, 12, MyMatrix)
2. length(which(MyMatrix>3 & MyMatrix<8))
3. MyMatrix[,12] <- as.character(MyMatrix[,12])

> newMatrix <- data.frame(MyMatrix, RowTest = ifelse(rowSums(MyMatrix)>70, TRUE,

+ ifelse(rowSums(MyMatrix)<70, FALSE, NA)))

**Problem Set 3 (Advanced)**

**Section 1**

> test <- function(x) {

+ return(“Hello world.”)

+ }

**Section 2**

1. Abbreviated sepal length with SL, sepal width with SW, etc.

> subiris <- function(x) {

+ df1 <- data.frame(subset(x, Species=="setosa"))

+ colnames(df1) <- c("setosa.SL", "setosa.SW", "setosa.PL", "setosa.PW", "Species")

+ df2 <- data.frame(subset(x, Species=="versicolor"))

+ colnames(df2) <- c("versicolor.SL", "versicolor.SW", "versicolor.PL", "versicolor.PW", "Species")

+ df3 <- data.frame(subset(x, Species=="virginica"))

+ colnames(df3) <- c("virginica.SL", "virginica.SW", "virginica.PL", "virginica.PW", "Species")

+ df1$Species <- NULL; df2$Species <- NULL; df3$Species <- NULL

+ d <- data.frame(df1, df2, df3)

+ return(d)

+ }

> attach(iris)

> sortiris <- function(x) {

+ s <- ifelse(Sepal.Width>3.1, Sepal.Length+Petal.Length, ifelse(Sepal.Width<3.1,

+ Sepal.Length-Petal.Length, NA))

+ return(s)

+ }

> library(dplyr)

> library(magrittr)

> avgbycyl <- function(x) {

+ car\_data <- mtcars %>% subset(cyl==x)

+ return(mean(car\_data$mpg))

+ }

> library(foreach)

> powersim <- function(num) {

+ a <- c(1:num)

+ b <- foreach(a=1:num, .combine=rbind) %do% sample(1:69, 5, replace=F)

+ c <- matrix(foreach(a=1:num) %do% sample(1:26, 1), ncol=1)

+ cbind(b,c)

+ }

Or

> powersim <- function(x) {

+ foreach(1:x, .combine=rbind) %do% sample(1:69, 5, replace=FALSE) %>%

+ data.frame(pow=matrix(foreach(1:x) %do% sample(1:26, 1)))

+ }

> powersimtest <- function(r1,r2,r3,r4,r5,p) {

+ a <- powersim(1000000)

+ b <- ifelse(a[1:1000000,]==c(r1,r2,r3,r4,r5,p), TRUE,

+ ifelse(a[1:1000000,]!=c(r1,r2,r3,r4,r5,p), FALSE, NA))

+ b <- ifelse(b==TRUE, 1, ifelse(b==FALSE, 0, NA))

+ b1 <- rowSums(b)

+ b <- matrix(data=ifelse(b1==6, TRUE, ifelse(b1!=6, FALSE, NA)), ncol=1)

+ return(which(b))

+ }